RESEARCH PAPER

Fuzzy-Constrained Graph Pattern Matching in Medical Knowledge Graphs

Lei Li[1,2,3†], Xun Du[3], Zan Zhang[3] & Zhenchao Tao[4]

[1]Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Hefei University of Technology, Hefei 230601, China

[2]Intelligent Interconnected Systems Laboratory of Anhui Province (Hefei University of Technology), Hefei 230601, China

[3]School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China

[4]The First Affiliated Hospital of University of Science and Technology of China, Hefei 230031, China

Keywords: Graph pattern matching; Medical Knowledge Graphs; Fuzzy constraints; Breast cancer; Diagnostic classification

Citation: Li, L. et al.: Fuzzy-Constrained Graph Pattern Matching in Medical Knowledge Graphs. Data Intelligence 4(3), 599-619 (2022). DOI: 10.1162/dint_a_00153

Received: Oct. 10, 2021; Revised: Jan. 15, 2022; Accepted: Apr. 10, 2022

ABSTRACT

The research on graph pattern matching (GPM) has attracted a lot of attention. However, most of the research has focused on complex networks, and there are few researches on GPM in the medical field. Hence, with GPM this paper is to make a breast cancer-oriented diagnosis before the surgery. Technically, this paper has firstly made a new definition of GPM, aiming to explore the GPM in the medical field, especially in Medical Knowledge Graphs (MKGs). Then, in the specific matching process, this paper introduces fuzzy calculation, and proposes a multi-threaded bidirectional routing exploration (M-TBRE) algorithm based on depth first search and a two-way routing matching algorithm based on multi-threading. In addition, fuzzy constraints are introduced in the M-TBRE algorithm, which leads to the Fuzzy-M-TBRE algorithm. The experimental results on the two datasets show that compared with existing algorithms, our proposed algorithm is more efficient and effective.

† Corresponding author: Lei Li (E-mail: lilei@hfut.edu.cn, ORCID: 0000-0002-5374-7293).

1. INTRODUCTION

As a basic data structure, graphs are widely used in a lot of applications. For example, as for object anomaly checking, objects can be represented by graphs, and then anomalies can be discovered with certain graph algorithm [1]. Meanwhile, in order to determine whether a user is interested in a certain webpage, the webpages can be converted into multiple graphs, and with the multiple graphs taken as a bag, the bag can be classified and judged [2]. As a popular graph-based technology, graph pattern matching (GPM) has attracted a lot of attentions. Specifically, given a pattern graph, finding subgraphs from the data graph with a similar or the same structure as the pattern graph is named as GPM. As the research field of GPM has changed from the initial protein isomorphism [3, 4] to community detection [5, 6], expert discovery [7], recommendation systems [8], the discovery of social groups [9–11] and the identification of criminal groups [12], the definition of graph pattern has also changed accordingly.

Technically, GPM is originally defined based on subgraph isomorphism. Given a data graph G_D and a pattern graph G_P as input, it will return whether it contains a subgraph, and whether this subgraph has exactly the same topological structure as G_P. For example, we can guess the properties of unknown proteins from the properties of known proteins through this matching [3, 4]. However, this traditional subgraph isomorphism is too strict. In order to extend the application scenarios of GPM, Fan et al. [12] propose a bounded simulation, which extends the edge-to-edge matching to the edge-to-finite length path matching. However, this matching still does not make use of the rich attribute information on vertices and edges. Therefore, Liu et al. [13] propose a multi-constrained graph pattern matching (MC-GPM) problem to obtain more effective matching results. Afterwards, Liu et al. [14] propose a multiple fuzzy constrained graph pattern matching (MFC-GPM) based on MC-GPM, considering that some attributes do not require exact matching. However, the current application scenarios of GPM are mostly concentrated on complex networks, and there are very few research applications of GPM in the medical field, especially in Medical Knowledge Graphs.

Nowadays, the incidence of breast cancer is getting higher and higher, and the age is getting younger and younger. Breast cancer can be divided into ductal carcinoma in situ, lobular carcinoma in situ, invasive ductal carcinoma, invasive lobular carcinoma, and so on. Each type of breast cancer can be divided according to the primary tumor staging, regional lymph node staging, and distant metastasis staging. The purpose of this paper is to make a diagnosis through GPM technology before the patient's condition is diagnosed with surgery.

In this paper, to introduce GPM into the medical field, we propose the problem of GPM in MKGs and give relevant definitions. In addition, the M-TBRE algorithm is proposed, which firstly divides the pattern graph into pattern subgraphs, then obtains the matching results of the pattern subgraphs, and finally merges the matching results of the pattern subgraphs. M-TBRE can give the diagnosis distribution of the pattern graph, and return the best k diagnosis classification results according to the frequency of each diagnosis classification. Fuzzy constraints are also introduced in the M-TBRE algorithm, which extend it to the Fuzzy-M-TBRE algorithm, and the effectiveness of our algorithm are verified on two public data sets.

The rest of this paper is organized as follows. The related work of GPM is reviewed in Section 2. Then in Section 3, the concept of pattern matching in MKGs is introduced. Section 4 proposes a multi-threaded bidirectional routing exploration algorithm and a Fuzzy-M-TBRE algorithm to process GPM in MKGs. Section 5 introduces the data sets and conducts experiments to verify our proposed Fuzzy-M-TBRE algorithm, and Section 6 concludes our work in this paper.

2. RELATED WORK

According to the judgment based on bijective function or based on binary relationship, the research on GPM can be divided into isomorphism-based GPM and simulation-based GPM.

2.1 Isomorphism-Based GPM

Isomorphism-based GPM has a bijective function between the pattern graph and the data graph, and the topological structure of the matched data subgraph and pattern graph must be the same. Ullmann [15] first proposes a matching algorithm based on depth-first search. Cordella et al. [16] improve Ullmann's algorithm in terms of matching order and pruning strategy, and propose the VF2 algorithm. Tong et al. [17] propose the G-Ray method, which uses the goodness function to measure the degree of matching between a subgraph and the pattern graph, so that the optimal-*k* subgraphs can be returned. In addition, Cheng et al. [18] also propose a top-*k* matching algorithm, which sorts the matched subgraphs obtained based on the number of spanning trees, thereby returning the optimal-*k* subgraphs. Cheng et al. [19] propose the R-join algorithm based on the join index of the clustering graph and optimize the algorithm. Other representative algorithms include DDST algorithm [20], IncBMatch algorithm [21], and so on. Generally, as an NP-complete problem, Isomorphism-based GPM uses indexing [22,23] and parallel distributed [24–26] to improve the efficiency of matching.

Isomorphism-based GPM is mostly used in fields with strict structural requirements such as protein isomorphism, 3D object matching [27] and network abnormal behavior detection [28]. However, such matching is too strict for applications such as social networks or knowledge graphs that do not require strict matching accuracy. Therefore, simulation-based GPM research has emerged.

2.2 Simulation-Based GPM

As judged through binary relations, graph simulation is introduced by Henzinger et al. [29], but it is still an edge-to-edge matching, which cannot meet the requirements of many applications. Fan et al. [12] extend the graph simulation and propose a bounded simulation, where the edge of the pattern graph can be matched to a path, and the length of this path does not exceed the given constraint value *k*. Based on the bounded simulation, Ma et al. [30] propose a strong simulation, which can well preserve the topological structure of the pattern graph. There is a lot of attribute information on vertexes and edges in big graph data, but these existing work does not consider this information. Liu et al. [13] consider this information to extend the bounded simulation to MC-GPM and propose a baseline algorithm based on exploration and

a heuristic algorithm based on data graph compression index (HAMC). Since the HAMC algorithm only considers the constraint conditions of the matching path, which does not consider the minimization of the matching path length and the HAMC algorithm does not support a distributed computing structure, Liu et al. [31] propose an M-HAMC algorithm. Considering that the attribute values on vertexes and edges sometimes do not need to be exactly matched, on the basis of MC-GPM, Liu et al. [14] propose an MFC-GPM and an ETOF-K algorithm, which improves matching efficiency from two aspects: edge matching and edge connection. Based on the topologically ordered sequence of pattern graph vertexes, Liu et al. [32] propose the NTSS algorithm and optimize the algorithm by introducing two measures: caching mechanism and reverse edge matching. The caching mechanism solves the problem of repeated calculation of the same candidate path in multiple matching subgraphs, and the reverse edge matching prunes the candidate set of the edge with a partial degree of entry 0 in advance.

3. GRAPH PATTERN MATCHING

GPM is to find all data subgraphs that satisfy the pattern graph G_P in a given data graph G_D. In this section, we will give the relevant definitions of data graphs, pattern graphs, and graph pattern matching in MKGs.

3.1 Data Graph and Pattern Graph

The related definitions of the data graph and the pattern graph are as follows.

3.1.1 Data Graph

A data graph $G_D = (V, E, f_V^D, f_E^D)$ is a directed graph with vertex attributes and edge attributes, where

- V is the set of vertices of the data graph;
- E is the set of edges of the data graph, and $(v_i, v_j) \in E$ represents the directed edge from vertex $v_i \in V$ to vertex $v_j \in V$;
  - f_V^D is a function defined on V. $\forall v \in V$, $f_V^D(v)$ is the attribute set of v. In an MKG, each vertex v has a label ρ_r, and ρ_r represents the type of this vertex. The value of ρ_r is different, and the other attributes in the attribute set $f_V^D(v)$ are also different. The value of ρ_r can be DI, BI, MI, GW, OC, AL and PD;
  - f_E^D is the function defined on E. $\forall e \in E$, $f_E^D(v_i, v_j)$ is the attribute set of e. In an MKG, for a directed edge (v_i, v_j), $f_E^D(v_i, v_j)$, only contains $\rho_{v_i v_j}^{\text{pids}}$. $\rho_{v_i v_j}^{\text{pids}}$ is a list that stores patient numbers, that is, the identity information of vertex which comes from those patients;

DI: When the value of ρ_r is DI, the attribute set $f_V^D(v)$ of vertex v describes the diagnostic classification information of breast cancer, which includes pathological information ρ^h, T staging stage ρ^s, tumor length ρ^{TS}, the description of regional lymph nodes N staging stage ρ^{LN}, and M staging stage ρ^{DM} describing distant metastasis. The value of ρ^h can be 0, 1, 2, and 3 respectively representing "invasive ductal carcinoma",

"invasive lobular carcinoma", "ductal carcinoma in situ", and "lobular carcinoma in situ"; the value of ρ^{s} can be 0, 1, 2, 3, and 4; ρ^{TS} is a floating point number in cm. The value of ρ^{LN} can be N0, N1, N2, and N3; the value of ρ^{DM} can be M0, and M1.

BI: When the value of ρ_r is BI, the attribute set $f_V^D(v)$ of vertex v describes the basic information of the patient, which includes ρ^{CN}, ρ^{CP} and ρ^{age}. ρ^{CN} indicates whether the patient currently needs care, and its value is true or false; ρ^{CP} indicates that the patient is currently pregnant, and its value is true or false; ρ^{age} indicates the current age of the patient, and its value is a positive integer.

MI: When the value of ρ_r is MI, the attribute set $f_V^D(v)$ of vertex v describes the patient's menopausal information, and it only contains ρ^{MS}. The value of ρ^{MS} can be 0, 1, and 2, respectively, indicating pre-menopausal, perimenopausal, and post-menopausal.

GW: When the value of ρ_r is GW, the attribute set $f_V^D(v)$ of vertex v describes the patient's current overall well-being, and it only contains ρ^{GWB}. The value of ρ^{GWB} can be 0, 1, 2, 3, and 4, respectively, which means "fully active, no complaints or symptoms", "doing normal activities requires a little effort", "occasionally need help, but can meet most of the personal needs", "needs a lot of assistance and frequent medical care", "completely disabled, can only lie in a bed or a chair."

OC: When the value of ρ_r is OC, the attribute set $f_V^D(v)$ of vertex v describes whether the patient has cancers other than breast cancer, which includes ρ^{OCS} and ρ^{OCN}. When the value of ρ^{OCS} is false, the value of ρ^{OCN} is none; when the value of ρ^{OCS} is true, the value of ρ^{OCN} is the names of the patient's other cancers;

AL: When the value of ρ_r is AL, the attribute set $f_V^D(v)$ of vertex v describes the patient's axillary lymph nodes, which includes ρ^{LN}, ρ^{LS}, ρ^{IN}, ρ^{SN} and ρ^{CW}. The value of ρ^{LS} is true or false, indicating whether the patient's axillary lymph nodes are normal or not. The value of ρ^{IN} is true or false, indicating whether the supraclavicular lymph nodes of the patient are normal or not. The value of ρ^{SN} is true or false, indicating whether the subclavian lymph nodes of the patient are normal or not. The value of ρ^{CW} is true or false, indicating whether the patient's chest wall is normal or not. The value of ρ^{LN} is a positive integer, which means that several of the three of the patient's supraclavicular lymph node, subclavian lymph node, and chest wall have problems.

PD: When the value of ρ_r is PD, the attribute set of vertex v describes some diagnosis information of the patient in the past, which includes ρ^{aid}, ρ^{ane}, ρ^{aut}, ρ^{lun}, ρ^{dia}, ρ^{car}, ρ^{ost} and ρ^{rep}. The value of ρ^{aid} is true or false, indicating whether the patient has AIDS; the value of ρ^{ane} is true or false, indicating whether the patient has anemia; the value of ρ^{aut} is true or false, indicating whether the patient has autoimmune disease; The value of ρ^{lun} is true or false, indicating whether the patient has lung cancer; the value of ρ^{dia} is true or false, indicating whether the patient has diabetes; the value of ρ^{car} is true or false, indicating whether the patient has cardiovascular disease; The value of ρ^{ost} is true or false, which indicates whether the patient has osteoporosis; the value of ρ^{rep} is true or false, which indicates whether the patient's reproductive organs are diseased.

3.1.2 Pattern Graph

A pattern graph $G_P = (V_p, E_P, f_V^P, f_E^P, f_l^P, f_m^P)$ is a directed graph with vertex attributes and edge attributes, where:

- V_p is the set of vertices of the pattern graph;
- E_p is the set of edges of the pattern graph, and $(u_i, u_j) \in E_P$ represents the directed edge from vertex $u_i \in V_P$ to vertex $u_j \in V_P$;
- f_V^P is a function defined on V_p, and $\forall v \in V_P$, $f_V^P(v)$ is the attribute set of v. In an MKG, the function $f_V^P(u)$ corresponding to the vertex u has the same meaning as the attribute set of the above vertex in the data graph.
- f_E^P is the function defined on E_p. $\forall e \in E_P$, $f_E^P(e)$ is the attribute set of e. In an MKG, for a directed edge (u_i, u_j), $f_E^P(u_i, u_j)$ only contains $\rho_{u_i u_j}^{\text{pids}}$. $\rho_{u_i u_j}^{\text{pids}}$ is a list that stores patient numbers, that is, the identity information of the vertices comes from those patients.
- f_l^P is the function defined on E_p. $\forall (u_i, u_j) \in E_p$, $f_l^P(u_i, u_j)$ is the length constraint of the edge (u_i, u_j), and its value is a positive integer k or a symbol *, respectively, indicating that the length of the interval from v_i to v_j does not exceed k, or there is no length limit. In an MKG, $f_l^P(u_i, u_j) = 1$;
- f_m^P is a set of membership constraint functions defined on vertex attributes and edge attributes.

3.1.3 Fuzzy constraints

During matching, it would be better to get more and better matching results. Because in the actual matching, each matched subgraph corresponds to a patient who has roughly the same health information as the patient to be diagnosed in the pattern graph. The more obtained matches, the better experience will be used for reference in the treatment of patients corresponding to the pattern graph. However, in practice, it is possible that a subgraph in a data graph can be well satisfied with other constraints, but because some less important attribute constraints on a vertex cannot be satisfied, the subgraph cannot become a matching result. In addition, some attribute constraints on vertexes do not need to be accurately matched when matching, and their differences only need to fall within a certain range. Therefore, we introduce fuzzy constraints to GPM in MKGs.

In the MKG, the membership function $f_m^P = \{f_{\text{age}}^m\}$ is only considered to introduce a fuzzy constraint to the age attribute. f_{age}^m represents the membership function defined on the vertex age attribute ρ_{age}. The constraint value of f_{age}^m is set to 3. The membership function f_{age}^m is defined as Eq. (1), where *abs* is the absolute value function, ρ_v^{age} represents the age attribute constraint value of vertex v in pattern graph G_P. ρ_u^{age} represents the age attribute constraint value of vertex u in data graph G_D. During matching, age attribute ρ_{age} only needs to satisfy $f_{\text{age}}^m \le 3$.

$$f_{age}^m = \text{abs}(\rho_u^{\text{age}} - \rho_v^{\text{age}}) \tag{1}$$

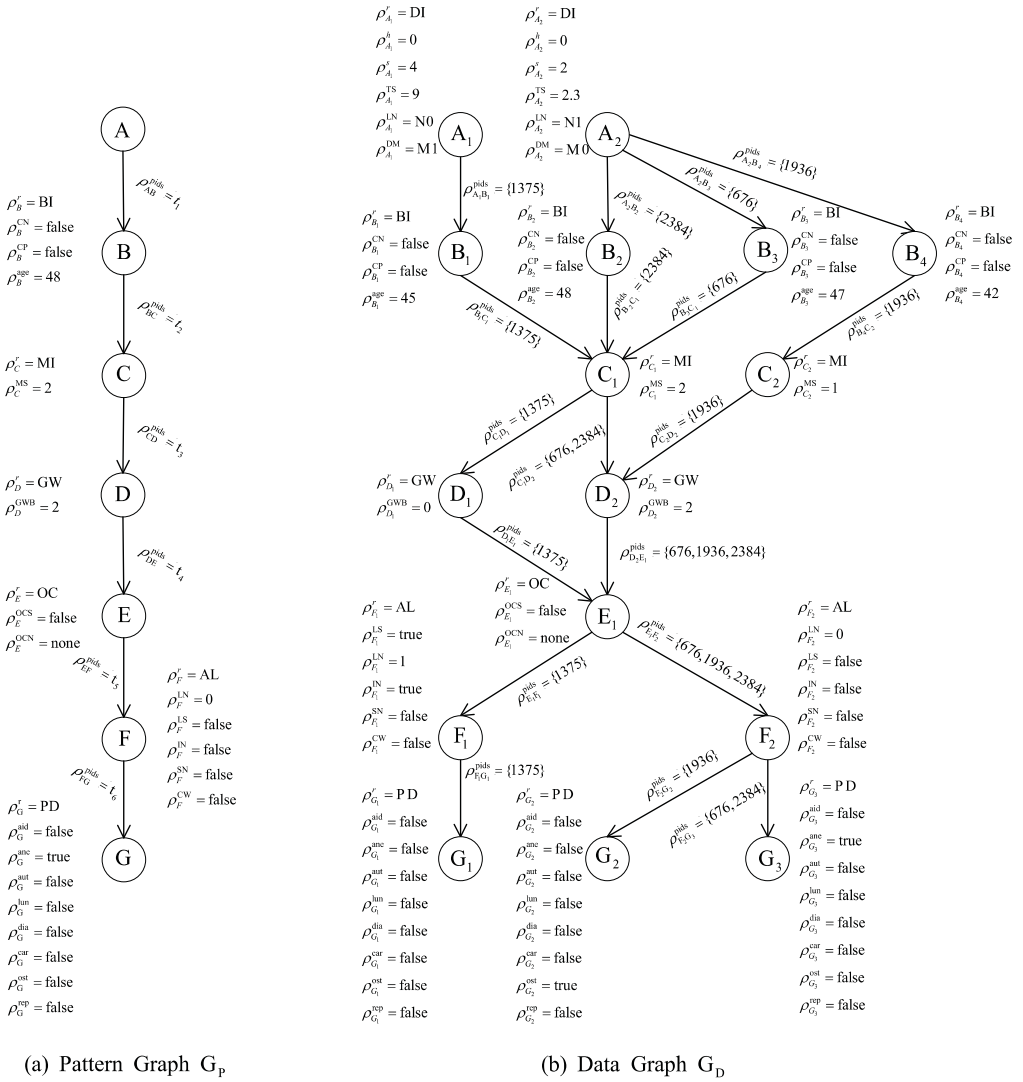
3.2 Pattern Matching

The matched subgraph $G_{sub} = (V_{sub}, E_{sub}, f^D_{V_{sub}}, f^D_{e_{sub}})$ is a subgraph of the data graph G_D and matches the pattern graph G_P. The number of matched subgraphs may not be unique, where $G_{sub} \subset G_D$, $V_{sub} \subset V$, $E_{sub} \subset E$, $f^D_{V_{sub}} \subset f^D_V$, $f^D_{E_{sub}} \subset f^D_E$; The definition of pattern matching in the MKG is as follows.

For a pattern graph $G_P = (V_p, E_P, f^P_V, f^P_E, f^P_l, f^P_m)$ and a data graph $G_D = (V, E, f^D_V, f^D_E)$, G_D matches G_P, denoted as $G_P \trianglelefteq G_D$, if there is a binary relationship:

- for all $u \in V_P$, there is $v \in V$ such that $(u, v) \in S$, which means that there is a vertex v in V that matches u, that is, v satisfies $f^P_V(u)$. If age attribute $\rho^{age}_{u_i}$ is included, $f^P_m = \{f^m_{age}\}$ represents the membership function defined on the age attribute of u, then ρ^{age}_u only needs to satisfy $f^m_{age} \leq 3$. Except for the age attribute ρ^{age}_u, the values corresponding to the other attributes of v must be equal to the values of the attributes corresponding to u before it can be determined that v_i matches u_i.
- for each pair $(u_i, v_i) \in S$,
  * $u_i \sim v_i$, and
  * for each edge (u_i, u_j) in E_P, there is a path from v_i to v_j in G_D such that $(u_j, v_j) \in S$. Because of $f^P_l(u_i, u_j) = 1$, this path can be regarded as the edge from v_i to v_j in G_D;

Example 1: As shown in Figure 1, G_D is a data graph composed of related information of multiple breast cancer patients. The attribute information of some vertexes contained in the data graph saves the diagnostic classification information of breast cancer. In the data graph G_D, each vertex represents some information of the patient. For the function $f_E^D(A_1, B_1)$ defined on the directed edge (A_1, B_1) in G_D, $f_E^D(A_1, B_1)$ only contains the attribute $\rho^{pids}_{A_1B_1}$. For example, the value of $\rho^{pids}_{A_1B_1}$ is 1375, which means that the relevant information on the B_1 vertex comes from the breast cancer patient numbered 1375. The pattern graph G_P is the health status of a patient to be diagnosed. The vertices *B, C, D, E, F*, and *G* respectively represent the patient's basic information, menopausal status, general well-being, information on cancers other than breast cancer, axillary lymph nodes and information about past diagnoses. Vertex *A* is the diagnostic information of this patient, but it is unknown and needs to be obtained through GPM. Since all vertex information in the pattern graph comes from the same patient, we need to find a patient number as the attribute constraint information on the edges to get the matching result of the pattern graph.



(a) Pattern Graph G_P (b) Data Graph G_D

Figure 1. Data graph and pattern graph in an MKG

Example 2: As shown in Figure 1, it is easy to find a subgraph M_{sub1} from data graph G_D that matches pattern graph G_P. M_{sub1} passes through vertexes A_2, B_2, C_1, D_2, E_1, F_2 and G_3. Vertex A_2 is the breast cancer diagnosis result of the pattern graph G_P. The attribute constraint value on the edges in M_{sub1} is 2384, which means that the patient with the number 2384 is closer to the health status of the patient corresponding to G_P.

After introducing fuzzy constraints, since $f_{age}^m = \text{abs}(\rho_B^{age} - \rho_{B_3}^{age}) = 1$, it does not exceed the membership function constraint value 3 on the age attribute. In addition, $\rho_B^r = \rho_{B_3}^r, \rho_B^{CN} = \rho_{B_3}^{CN}, \rho_B^{CP} = \rho_{B_3}^{CP}$, and vertex B

matches vertex B_3. We can get a new matched subgraph M_{sub2} that passes through vertexes A_2, B_3, C_1, D_2, E_1, F_2 and G_3. The attribute constraint value on the edges in M_{sub2} is 676.

4. GRAPH PATTERN MATCHING IN MEDICAL KNOWLEDGE GRAPHS

In this section, we propose a multi-threaded bidirectional routing exploration algorithm M-TBRE to solve the GPM problem in MKGs.

4.1 Algorithm Description

The emergence of multi-core CPU can realize the parallel processing of tasks and speeds up the execution of programs. Since the multi-constrained GPM problem is an NP-complete problem, in order to speed up the matching speed and return the matched results quickly, here we consider adopting multi-threading to solve this GPM problem. In the matching process, the idea of divide and conquer is adopted. For a pattern graph G_P, it can be divided into several pattern subgraphs. After the matching of each pattern subgraph is completed in the data graph G_D, the matched results of each pattern subgraph can be connected to obtaining the matched results of the pattern graph G_P. The matching of pattern subgraphs can be delivered as subtasks to multiple threads to complete independently, so that matching results can be obtained quickly.

4.2 Algorithm Flow

In the M-TBRE algorithm, since the pattern graph of the MKG can be regarded as a path, we can segment the pattern graph according to the intermediate vertexes of this path, divide the pattern graph into two parts, and obtain two pattern subgraphs. Next, to match the two pattern subgraphs, the matched results are connected to obtaining the matched results of the pattern graph.

The detailed steps of the M-TBRE algorithm are shown in Algorithm 1. First, the intermediate vertex V_P^{mid} of the pattern graph G_P and the candidate vertex set cand_{mid} of V_P^{mid} need to be obtained, as shown in lines 1–2. In line 3, *pool* and *tempInfo* represent the thread pool and the temporary result of the matching, respectively. The number of threads in *pool* can be set according to the actual situation. Then the pattern graph G_P is divided into two sub-pattern graphs G_P^{sub1} and G_P^{sub2} with intermediate vertex V_P^{mid} as the dividing point, and the two sub-pattern graphs are matched in the data graph G_D. Traversing the candidate vertex set cand_{mid} of V_P^{mid} to complete the matching of the sub-pattern graphs G_P^{sub1} and G_P^{sub2}, as shown in lines 4–27. For each element $\text{cand}_{\text{mid}}[i]$ in cand_{mid}, we use the attribute constraint $\rho_{\text{pe}}^{\text{pids}}$ on each forward edge e_D^{pe} of $\text{cand}_{\text{mid}}[i]$ to intersect the attribute constraint $\rho_{\text{ae}}^{\text{pids}}$ on each successor edge e_D^{ae} to obtain ρ^{pids}, which saves the common patient number information of the current forward edge e_D^{pe} and the current successor edge e_D^{ae}, as shown in lines 6–25. For each patient number ρ^{pid} in ρ^{pids}, ρ^{pid} is taken as the attribute constraint on the edge to complete the matching of G_P^{sub1} and G_P^{sub2}, as shown in lines 14–21. *tempInfo* stores the partial matched result with ρ^{pid} as the edge attribute constraint, as shown in lines 15–16. The thread pool submits subtasks MC-SEM and MC-FEM to complete the matching of G_P^{sub1} and G_P^{sub2}

respectively, as shown in lines 19–20. The algorithm RM merges the matched results, as shown in line 28. The MC-SEM algorithm can complete the matching of the pattern subgraph G_P^{sub1}, where v_P^{curr}, v_D^{curr}, ρ^{pid} and *tempInfo* respectively represent the pattern vertex to be matched, the candidate vertex of the pattern vertex v_P^{curr} to be matched, the attribute constraint value (patient number) of the edge, and the temporary result of the matching. If vertex v_D^{curr} matches vertex v_P^{curr} but v_P^{curr} does not have a successor edge, that is, when the out-degree of v_P^{curr} is 0, the matching of the pattern subgraph G_P^{sub1} is completed, and the matched result when ρ^{pid} is used as the attribute constraint on the edge is saved in *tempInfo*, such as Algorithm 2 is shown in lines 2–7. If vertex v_D^{curr} matches vertex v_P^{curr} and v_P^{curr} has a successor edge, then traverse the successor edge e_D^{ae} of v_D^{curr}. When the attribute constraint ρ_{ae}^{pids} on e_D^{ae} includes ρ^{pid}, the matching of pattern vertex e_P^{ae}. tailNode is recursively completed, as shown in lines 8–16.

Algorithm 1 M-TBRE Algorithm

Input: G_P,G_D
Output: result set M_{sub}

1: Get the intermediate node of the pattern graph G_P,v_P^{mid}
2: Get the candidate node set of N_{mid} from V,$cand_{mid}$
3: **initialize**: pool, tempInfo
4: i = 0
5: **while** i < $cand_{mid}$.length **do**
6: $v_{cand}^{curr} = cand_{mid}$ [i]
7: $e_D^{pe} = v_{cand}^{curr}$.firstPreEdge
8: **while** $e_D^{pe} \neq$ NULL **do**
9: $\rho_{pe}^{pids} = e_D^{pe}$.pids
10: $e_D^{ae} = v_{cand}^{curr}$.firstAfterEdge
11: **while** $e_D^{ae} \neq$ NULL **do**
12: $\rho_{ae}^{pids} = e_D^{ae}$.pids
13: $\rho^{pids} = \rho_{ae}^{pids} \cap \rho_{pe}^{pids}$
14: **for** $\rho^{pid} \in \rho^{pids}$ **do**
15: **initialize**: state [3]
16: tempInfo.put (ρ^{pid}, state)
17: $e_P^{pe} = N_{mid}$.firstPreEdge
18: $e_P^{pe} = N_{mid}$.firstAfterEdge
19: pool.submit (MC-SEM, e_P^{ae}.tailNode, e_D^{ae}.tailNode, ρ^{pid}, tempInfo)
20: pool.submit (MC-FEM, e_P^{pe}.tailNode, e_D^{pe}.tailNode, ρ^{pid}, tempInfo)
21: **end for**
22: $e_D^{ae} = e_D^{ae}$.nextEdge
23: **end while**
24: $e_D^{pe} = e_D^{pe}$.nextEdge
25: **end while**
26: i++
27: **end while**
28: M_{sub} = RM (tempInfo)
29: return M_{sub}

Algorithm 2 Multi-Constrained Subsequent Edge Matching,MC-SEM

Input: v_P^{curr},v_D^{curr},ρ^{pid},tempInfo
1: **if** v_D^{curr}.equals (v_P^{curr}) **then**
2: $e_P^{ae} = v_P^{curr}$.firstAfterEdge
3: **if** e_P^{ae} == NULL **then**
4: state [] = tempInfo.get(ρ^{pid})
5: state [1] = 1
6: return
7: **end if**
8: $e_D^{ae} = v_D^{curr}$.firstAfterEdge
9: **while** $e_D^{ae} \neq$ NULL **do**
10: $\rho_{ae}^{pids} = e_D^{ae}$.pids
11: **if** $\rho^{pid} \in \rho_{ae}^{pids}$ **then**
12: MC-SEM $(e_P^{ae}$.tailNode, e_D^{ae}.tailNode, ρ^{pid}, tempInfo)
13: return
14: **end if**
15: $e_D^{ae} = e_D^{ae}$.nextEdge
16: **end while**
17: **end if**

The MC-FEM algorithm can complete the matching of the pattern subgraph G_P^{sub2}. The processing process of the MC-FEM algorithm is similar to that of the MC-SEM algorithm, except that MC-FEM completes the matching of the pattern subgraph G_P^{sub2} according to the reverse depth-first search strategy.

Algorithm 3 Multi-Constrained Forward Edge Matching,MC-FEM

Input: v_P^{curr},v_D^{curr},ρ^{pid},tempInfo
1: $e_P^{pe} = v_P^{curr}$.firstPreEdge
2: **if** e_P^{pe} == NULL **then**
3: state [] = tempInfo.get (ρ^{pid})
4: state [0] = 1
5: Save the attributes of v_D^{curr} in state [2]
6: return
7: **end if**
8: **if** v_D^{curr}.equals (v_P^{curr}) **then**
9: $e_D^{pe} = v_D^{curr}$.firstPreEdge
10: **while** $e_D^{pe} \neq$ NULL **do**
11: $\rho_{pe}^{pids} = e_D^{pe}$.pids
12: **if** $\rho^{pid} \in \rho_{pe}^{pids}$ **then**
13: MC-FEM $(e_P^{pe}$.tailNode, e_D^{pe}.tailNode, ρ^{pid}, tempInfo)
14: return
15: **end if**
16: $e_D^{pe} = e_D^{pe}$.nextEdge
17: **end while**
18: **end if**

The RM algorithm can complete the connection operation of the matching results of pattern subgraphs G_P^{sub1} and G_P^{sub2}. When a given value is used as an attribute constraint on all edges, and the flag bits representing the matching results of G_P^{sub1} and G_P^{sub2} are both 1, then combining the matching results of G_P^{sub1} and G_P^{sub2} is a matching result of the pattern graph G_P, such as lines 4–6 in Algorithm 4.

Algorithm 4 Result Merge,RM

Input: tempInfo
Output: result set M_{res}

```
1: for entry ∈ tempInfo do
2:     ρ^pid = entry.key
3:     state = entry.value
4:     if state [0] == 1 and state [1] == 1 then
5:         state [2] .add(pid)
6:         M_res.add (state [2])
7:     end if
8: end for
9: return M_res
```

Example 3: In this example, G_P and G_D in Figure 1 are the pattern graph and the data graph, respectively. First, to obtain intermediate vertex D of pattern graph G_P and candidate vertex set $cand_{mid} = \{D_2\}$ of D. The pattern graph G_P is divided into pattern subgraph G_P^{sub1} which passes through vertexes A, B and C, and pattern subgraph G_P^{sub2} which passes through vertexes E, F and G. The forward edge (C_1, D_2) and the subsequent edge (D_2, E_1) of D_2 have the same attribute constraint $\rho^{pids} = \{676, 2384\}$. Taking the matching of G_P^{sub2} as an example, the attribute constraint $\rho^{pids}_{C_1D_2} = \{676, 2384\}$ on edge (C_1, D_2) contains $\rho^{pid} = 2384$, and C_1 matches C at the same time, so edge $(C_1, D_2, G_D) \simeq (C, D, G_P)$. We can get $(C_1, D_2, G_D) \simeq (C, D, G_P)$ and $(A_2, B_2, G_D) \simeq (A, B, G_P)$. The matching of G_P^{sub2} takes $\rho^{pid} = 2384$ as the attribute constraint, and the attribute constraint information of vertex A_2 is the diagnosis classification result of G_P^{sub2}. In the same way, we can get the matched results $(D_2, E_1, G_D) \simeq (D, E, G_P)$, $(E_1, F_2, G_D) \simeq (E, F, G_P)$, $(F_2, G_3, G_D) \simeq (F, G, G_P)$ of G_P^{sub1} when $\rho^{pid} = 2384$ is the attribute constraint on edge E_P. Both G_P^{sub1} and G_P^{sub2} have matched results, so the diagnostic classification information of G_P is the diagnostic classification information of G_P^{sub1}, and the patient number in G_D is 2384. Finally, two matching subgraphs are obtained through the M-TBRE algorithm. $M_{sub1} = \{V_m, E_m, f_v, f_e\}$, where $V_M = \{A_2, B_2, C_1, D_2, E_1, F_2, G_3\}$, $f_e = \{2384\}$ and $E_M = \{(A_2, B_2), (B_2, C_1), (C_1, D_2), (D_2, E_1), (E_1, F_2), (F_2, G_3)\}$. $M_{sub2} = \{V_m, E_m, f_v, f_e\}$, where $V_M = \{A_2, B_3, C_1, D_2, E_1, F_2, G_3\}$, $f_e = \{676\}$ and $E_M = \{(A_2, B_3), (B_3, C_1), (C_1, D_2), (D_2, E_1), (E_1, F_2), (F_2, G_3)\}$. The patients numbered 676 and 2384 can be used for reference when treating the patients corresponding to the pattern graph G_P.

5. EXPERIMENTS

In this section, we conduct experiments on two public MKGs. The details of these two datasets are shown in Table 1. We propose and implement the M-TBRE algorithm to complete the pattern matching of MKG. Since the M-TBRE algorithm divides the pattern graph into two sub-pattern graphs, for different edge

attribute constraint values ρ^{pid}, the matching of these two sub-pattern graphs is delivered to the thread pool as a subtask for execution. More or less the number of threads in the thread pool will affect the execution results of the algorithm. We will set a different number of threads to measure the efficiency dynamics of the M-TBRE algorithm. In addition, to obtain more matched results, we introduce the fuzzy constraint, which is a membership function for the age attribute constraint in the vertexes. The age attribute constraint on the data graph vertexes does not need to be the same as the attribute constraints in the pattern graph during matching, but only needs to go through the calculated result of the membership function and satisfy the corresponding membership constraint value. Together with the M-TBRE algorithm, we have the Fuzzy-M-TBRE algorithm. The Fuzzy-M-TBRE and M-TBRE algorithms can be compared to prove the effectiveness of the introduced fuzzy constraints.

Table 1. The detail information of two datasets.

Dataset	Vertices	Edges	Description
Female-breast-cancer-2013a	10812	20366	A graph about breast cancer patients
Breastcancer-femalepatient-2016A	101221	200845	A graph about breast cancer patients

5.1 Experimental Settings and Implementation

The MKG used in the experiment is about breast cancer. Dataset-1 and dataset-2 are used to represent the dataset Female-breast-cancer-2013a and the dataset Breastcancer-femalepatient-2016A, respectively. Dataset-1 is composed of the physical condition information of 10,000 breast cancer patients, and dataset-2 is composed of 100,000 breast cancer patients. In our experiment, several pattern graphs are used, but these pattern graphs are similar to the pattern graph shown in Figure 1. Our membership function is only for the age attribute of the vertex, and the membership constraint value is set to 3. Both M-TBRE and Fuzzy-M-TBRE are implemented using Java and running on a PC with Intel(R) Core(TM) i9-10900F CPU @2.81G GHz, 32 GB RAM and Windows 10 operating system.

5.2 Experimental Results and Analysis

5.2.1 Experiments on Execution Time

This experiment studies the execution time change when we set different thread numbers for the thread pool used in the M-TBRE algorithm, and the algorithm completes the GPM. To prevent error interference, the results in the experiment are the arithmetic mean after 10 runs.

As shown in Figure 2 and Figure 3, the abscissa represents different pattern graphs, and the ordinate represents the matching time of these pattern graphs. The M-TBRE-1 algorithm represents that the number of threads in the thread pool in the M-TBRE algorithm is set to 1. Since the pattern graph in the MKG is a path, and the edge join strategy proposed in the ETOF-K algorithm does not take effect in the matching, the performance of the ETOF-K algorithm and the M-TBRE-1 algorithm is almost the same on dataset-1 and dataset-2. The reverse matching strategy of the NTSS algorithm is invalid in the matching process, but its

caching mechanism avoids the double calculation of the same path, so the NTSS algorithm is better than the ETOF-K algorithm and the M-TBRE-1 algorithm on dataset-1 and dataset-2.

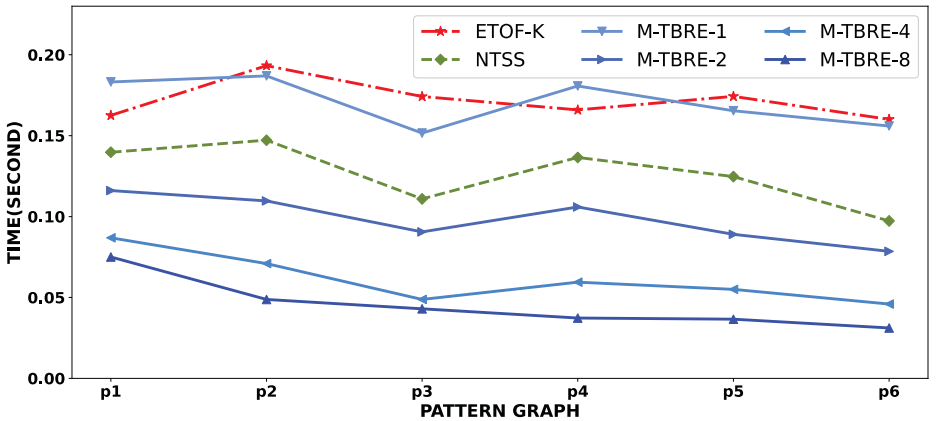


Figure 2. Matching time of different pattern graphs on Female-breast-cancer-2013a.

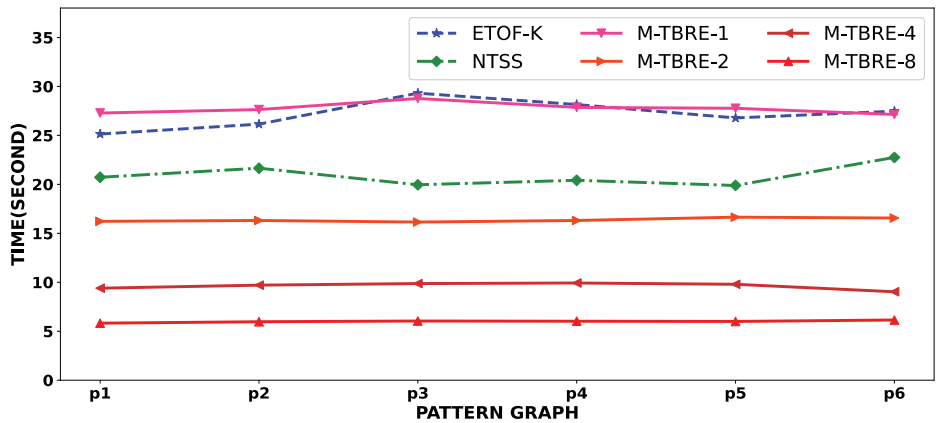


Figure 3. Matching time of different pattern graphs on Breastcancer-femalepatient-2016A.

However, our M-TBRE-1 algorithm can be extended to multithreaded algorithms, such as the M-TBRE-2 algorithm, M-TBRE-4 algorithm, M-TBRE-8 algorithm, which means that the number of threads in the thread pool is set to 2, 4, and 8, respectively. As can be seen from Figure 2 and Figure 3, the effect of the M-TBRE-2 algorithm has already exceeded the NTSS algorithm, which also proves the effectiveness of our proposed M-TBRE algorithm. In addition, Table 2 and Table 3 show the detailed execution time in seconds. Table 4 shows the comparison of the average execution time of these four algorithms on the two datasets. It can be seen that on the two data sets, as the number of threads increases, the execution time of the algorithm continues to decrease.

Table 2. Execution time on the Female-breast-cancer-2013a dataset.

	P1	P2	P3	P4	P5	P6	P7	P8
ETOF-K	0.1625	0.1933	0.1742	0.1659	0.1743	0.1601	0.1742	0.1633
NTSS	0.1398	0.1472	0.1109	0.1365	0.1247	0.0973	0.1148	0.1245
M-TBRE-1	0.1832	0.1869	0.1516	0.1807	0.1654	0.1560	0.1535	0.1700
M-TBRE-2	0.1161	0.1097	0.0905	0.1059	0.0890	0.0785	0.0789	0.0979
M-TBRE-4	0.0869	0.0709	0.0488	0.0595	0.0550	0.0459	0.0448	0.0512
M-TBRE-8	0.0750	0.0488	0.0430	0.0373	0.0366	0.0311	0.0294	0.0331

Table 3. Execution time on the Breastcancer-femalepatient-2016A dataset.

	P1	P2	P3	P4	P5	P6	P7	P8
ETOF-K	25.1406	26.1477	29.3101	28.1559	26.7859	27.4765	28.6518	27.4800
NTSS	20.7231	21.6519	19.9583	20.4127	19.8864	22.7456	19.7193	21.5986
M-TBRE-1	27.2807	27.6337	28.7748	27.8562	27.7634	27.1356	26.8715	27.3600
M-TBRE-2	16.2298	16.3104	16.1557	16.3067	16.6475	16.5609	16.4358	16.7567
M-TBRE-4	9.4020	9.7056	9.8746	9.9239	9.7943	9.0300	8.9316	8.9685
M-TBRE-8	5.8198	5.9732	6.0509	6.0256	5.9963	6.1483	6.1579	6.1895

Table 4. The comparison of execution time on two datasets.

Dataset	M-TBRE-1	M-TBRE-2	M-TBRE-4	M-TBRE-8	Percentage
Female-breast-cancer-2013a	0.1684	0.0958	—	—	43.11%
Female-breast-cancer-2013a	—	0.0958	0.0579	—	39.56%
Female-breast-cancer-2013a	—	—	0.0579	0.0418	27.81%
Breastcancer-femalepatient-2016A	27.5845	16.4254	—	—	40.45%
Breastcancer-femalepatient-2016A	—	16.4254	9.4538	—	42.44%
Breastcancer-femalepatient-2016A	—	—	9.4538	6.0452	36.06%

- For dataset-1, the execution time of the M-TBRE-2 algorithm is increased by 43.11% compared with the M-TBRE-1, and the execution time of the M-TBRE-4 algorithm is increased by 39.56% compared with the M-TBRE-2 algorithm, but compared with the M-TBRE-4 algorithm, the execution time of the M-TBRE-8 algorithm is only increased by 27.81%. This is because dataset-1 itself is small in scale, and the time spent on thread context switching and system state transitions occupies a large proportion of the total time.
- For dataset-2, its scale is larger, and the total execution time of the algorithm is also larger. The time spent on thread context switching and system state transition takes up a relatively small proportion of the total time. Therefore, M-TBRE-8 in dataset-2 still increased by 36.06%.

For our proposed M-TBRE algorithm, as the number of threads increases, the execution speed is also accelerating. But when the number of threads reaches a certain level, the increase in execution speed will slow down, as shown by M-TBRE-8 in Figure 2. If the dataset is larger, or when the M-TBRE algorithm submits more subtasks, this slowing downtrend will become slower. Compared with the M-TBRE-4 algorithm,

the M-TBRE-8 algorithm in dataset-1 increased by 27.81%, while in dataset-2, the M-TBRE-8 algorithm increased by 36.06% compared with the M-TBRE-4 algorithm.

5.2.2 Experiments on Fuzzy Constraints

This experiment studies the change in the number of matching subgraphs when our M-TBRE algorithm introduces fuzzy constraints. The Fuzzy-M-TBRE algorithm is the algorithm after M-TBRE introduces fuzzy constraints. Since our Fuzzy-M-TBRE algorithm can get all the matched results of the pattern graph, we can compare the changes in the total number of matches before and after the introduction of fuzzy constraints.

As shown in Figure 4 and Figure 5, the abscissa represents different pattern graphs, and the ordinate represents the number of matched subgraphs. On dataset-1 and dataset-2, for the same pattern graph, the Fuzzy-M-TBRE algorithm returns more matched results than the M-TBRE algorithm. Each matched subgraph corresponds to a breast cancer patient. When treating the patient corresponding to the pattern graph, please refer to the treatment plan of the corresponding patient in these matched subgraphs. Introducing fuzzy constraints can get more treatment options. Therefore, it is necessary to introduce fuzzy constraints into the M-TBRE algorithm.

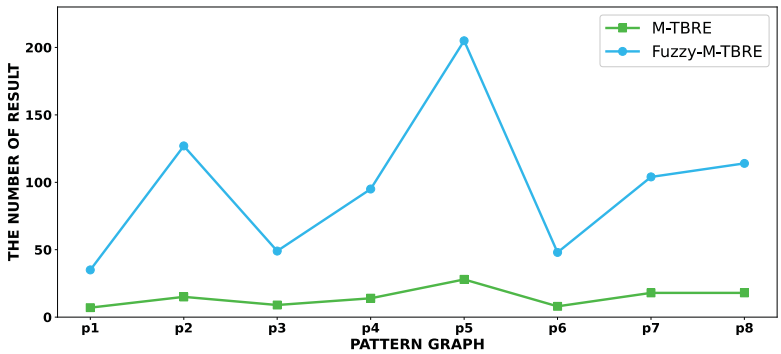


Figure 4. The number of matched subgraphs of different pattern graphs on Female-breast-cancer-2013a.

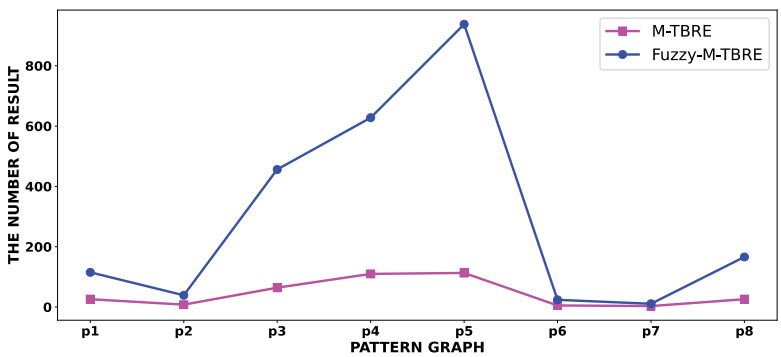


Figure 5. The number of matched subgraphs of different pattern graphs on Breastcancer-femalepatient-2016A.

6. CONCLUSION

In this paper, we put forward the problem of GPM in MKGs, and provide related definitions. In order to solve this problem, an M-TBRE algorithm is proposed, which divides the pattern graph into several pattern subgraphs, uses multi-threaded bidirectional routing to complete the matching of the pattern subgraphs, and then merges the matching results. In addition, fuzzy constraints are introduced to obtain more matching subgraphs. Each matched subgraph corresponds to a past patient. The patients corresponding to these matched subgraphs have the same physical condition as the patient corresponding to the pattern graph, so the treatment plan of the patients corresponding to these matched subgraphs can be used for reference in the treatment of the patient corresponding to the pattern graph. In this way, better and more effective treatment plans can be developed for patients corresponding to the pattern graph. We conduct verification experiments on the M-TBRE algorithm on two public MKG datasets. Experimental results show that our proposed M-TBRE algorithm has better performance. Furthermore, the necessity of introducing fuzzy constraints is also demonstrated, which leads to the outperformance of the Fuzzy-M-TBRE algorithm. In the future, we will further research and improve the M-TBRE algorithm, and study the dynamic graph pattern matching problem in MKGs oriented to the dynamics of pattern graph content.

ACKNOWLEDGMENTS

This work has been supported by the National Natural Science Foundation of China under grants 62076087 & 61906059 and the Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT) of the Ministry of Education of China under grant IRT17R32.

The first author would like to thank his wife Jun Zhang, his parents and friends during his fight with lung adenocarcinoma. “I leave no trace of wings in the air, but I am glad I have had my flight.”

AUTHOR CONTRIBUTIONS

All authors including L. Li (lilei@hfut.edu.cn), X. Du (dlx4339@163.com), Z. Zhang (zanzhang@hfut.edu.cn), and Z. Tao (zctao@ustc.edu.cn) took part in writing the paper. In addition, L. Li designed the algorithm and experiments, and provided the funding; X. Du designed and conducted experiments, and analyzed the data; Z. Tao analyzed the data.

REFERENCES

[1] Ma, X., Wu, J., Xue, S., et al.: A comprehensive survey on graph anomaly detection with deep learning. IEEE Transactions on Knowledge and Data Engineering (2021)

[2] Wu, J., Zhu, X., Zhang, C., et al.: Bag constrained structure pattern mining for multi-graph classification. IEEE Transactions on Knowledge and Data Engineering 26(10), 2382–2396 (2014)

[3] Hu, J., Ferguson, A.: Global graph matching using diffusion maps. Intelligent Data Analysis 20(3), 637–654 (2016)
[4] Tian, Y., Patel, J.: TALE: A tool for approximate large graph matching. In: Proceedings of the IEEE 24th International Conference on Data Engineering, pp. 963–972 (2018)
[5] Liu, F., Xue, S., Wu, J., et al.: Deep learning for community detection: progress, challenges and opportunities. In: Proceedings IJCAI, pp. 4981–4987 (2020)
[6] Su, X., Xue, S., Liu, F., et al.: A comprehensive survey on community detection with deep learning. IEEE Transactions on Neural Networks and Learning Systems, 1–21 (2021)
[7] Fan, W., Wang, X., Wu, Y.: Finding experts by graph pattern matching. In: Proceedings of the IEEE 29th International Conference on Data Engineering, pp. 1316–1319 (2008)
[8] Fan, W., Wang, X., Wu, Y.: Incremental graph pattern matching. ACM Transactions on Database Systems 38(3), 1–47 (2013)
[9] Khan, A., Golab, L., et al.: Compact group discovery in attributed graphs and social networks. Information Processing & Management 57(2), 102054 (2020)
[10] Ryota, S., Hitoshi, H., et al.: Social Group Discovery Extracting Useful Features using Multiple Instance Learning. Journal of Japan Society for Fuzzy Theory & Intelligent Informatics 28(6), 920–931 (2016)
[11] Chikhaoui, B., shimula, J., Wang, S.: Community Mining and Cross-Community Discovery in Online Social Networks. In: Proceedings of the International Conference on Network-Based Information Systems, pp. 176–187 (2020)
[12] Fan, W., Li, J., et al.: Graph pattern matching: from intractable to polynomial time. Proceedings of the VLDB Endowment 3(1–2), 264–275 (2010)
[13] Liu, G., Zheng, K., et al.: Multi-constrained graph pattern matching in large-scale contextual social graphs. In: Proceedings of the IEEE 31st International Conference on Data Engineering, pp. 351–362 (2015)
[14] Liu, G., Li, L., Wu, X.: Multi-fuzzy-constrained graph pattern matching with big graph data. Intelligent Data Analysis 24(4), 941–958 (2020)
[15] Ullmann, J.: An Algorithm for Subgraph Isomorphism. Journal of the ACM 23(1), 31–42 (1976)
[16] Cordella, L., Foggia, et al.: A (Sub) Graph Isomorphism Algorithm for Matching Large Graphs. IEEE transactions on pattern analysis and machine intelligence 26(10), 1367–1372 (2004)
[17] Tong, H., Faloutsos, C., et al.: Fast best-effort pattern matching in large attributed graphs. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 737–746 (2007)
[18] Cheng, J., Zeng, X., Yu, J.: Top-k graph pattern matching over large graphs. In: Proceedings of the IEEE 29th International Conference on Data Engineering, pp. 1033–1044 (2013)
[19] Cheng, J., Yu, J., et al.: Fast Graph Pattern Matching. In: Proceedings of the IEEE 24th International Conference on Data Engineering, pp. 913–922 (2008)
[20] Song, C., Ge, T., Chen, C., Wang, J.: Event pattern matching over graph streams. Proceedings of the VLDB Endowment 8(4), 413–424 (2014)
[21] Fan, W., Li, J., Luo, J., Tan, Z., Wang, X., Wu, X.: Incremental graph pattern matching. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 925–936 (2011)
[22] Yan, X., Yu, P., Han, J.: Graph indexing: a frequent structure-based approach. In: Proceedings of the 2004 ACM SIGMOD international conference on Management of data, pp. 335–346 (2004)
[23] Shasha, D., Wang, J., Giugno, R.: Algorithmics and applications of tree and graph searching. In: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 39–52 (2002)

[24] Afrati, F., Fotakis, D., Ullman, J.: Enumerating subgraph instances using map-reduce. In: Proceedings of the IEEE 29th International Conference on Data Engineering, pp. 62–73 (2013)
[25] Shao, Y., Cui, B., Chen, L., Ma, L., Yao, J., Xu, N.: Parallel subgraph listing in a large-scale graph. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, pp. 625–636 (2014)
[26] Huang, J., Venkatraman, K., Abadi, D.: Query optimization of distributed pattern matching. In: Proceedings of the 2014 IEEE 30th International Conference on Data Engineering, pp. 64–75 (2014)
[27] Demirci, M.: Graph-based shape indexing. Machine Vision and Applications 23(3), 541–555 (2012)
[28] Choudhury, S., Holder, L., Chin, G., Ray, A., Beus, S., Feo, J.: StreamWorks: a system for dynamic graph search. In: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, pp. 1101–1104 (2013)
[29] Henzinger, M., Henzinger, T., Kopke, P.: Computing simulations on finite and infinite graphs. In: Proceedings of the IEEE 36th Annual Foundations of Computer Science, pp. 453–462 (1995)
[30] Ma, S., Cao, Y., Fan, W., Huai, J., Wo, T.: Capturing topology in graph pattern matching. ACM Transactions on Database Systems 39(1), 4:1–4:46 (2014)
[31] Liu, G., Liu, Y., Zheng, K., Liu, A., Li, Z., Wang, Y., Zhou, X.: MCS-GPM: Multi-Constrained Simulation Based Graph Pattern Matching in Contextual Social Graphs. IEEE Transactions on Knowledge and Data Engineering 30(6), 1050–1064 (2018)
[32] Liu, G., Li, L., Liu, G., Wu, X.: Social Group Query Based on Multi-Fuzzy-Constrained Strong Simulation. Transactions on Knowledge Discovery from Data 16(3), 1–27 (2021)

AUTHOR BIOGRAPHY

Lei Li received his Bachelor's degree from Jilin University, Changchun, China, in 2004, his Master's degree from the Memorial University of Newfoundland, St. John's, Canada, in 2006, and his Ph.D. degree from Macquarie University, Sydney, Australia, in 2012. He is currently an Associate Professor at Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Intelligent Interconnected Systems Laboratory of Anhui Province (Hefei University of Technology), and School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China. His research interests include data mining, social computing and graph computing. He is a senior member of IEEE.
ORCID: 0000-0002-5374-7293

Xun Du received the Bachelor's degree in 2019, and Master's degree in 2022 from Hefei University of Technology, China. His research interests are in graph computing and social computing. Currently, he works as a software development engineer at Honor Terminals LTD.
ORCID: 0000-0002-7020-5883

Zan Zhang received his Ph.D. degree in Computer Science from Hefei University of Technology, China, in 2018. He is currently a lecturer at the Hefei University of Technology. His research interests include data mining and knowledge engineering.
ORCID: 0000-0002-6383-1683

Zhenchao Tao was born in Wuhu, Anhui, China, in 1985. He received the M. M. degree in oncology from Anhui Medical University, in 2012. From 2012 to 2015, he was a Resident, and from 2015 to 2017, he was an Attending Physician at the Department of Radiotherapy, Anhui Cancer Hospital. Since 2017, he has been an Attending Physician with the Department of Radiotherapy, The First Affiliated Hospital of University of Science and Technology of China. He published five articles as first author and participated in three provincial and national projects.

ORCID: 0000-0001-8142-9164